

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-212757

(43)Date of publication of application : 11.08.1995

(51)Int.Cl.

H04N 7/30

H04N 1/41

(21)Application number : 06-005789

(71)Applicant : TOSHIBA CORP  
TOSHIBA AVE CORP

(22)Date of filing : 24.01.1994

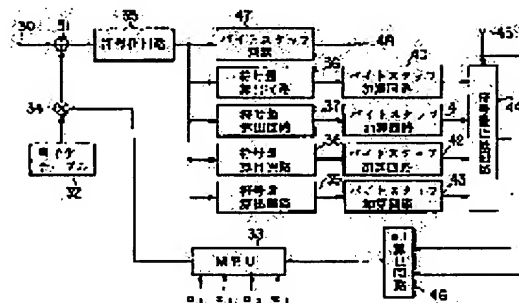
(72)Inventor : NAKAGAWA MASAKI  
HISATOMI SHUICHI

## (54) PICTURE COMPRESSION CODER

### (57)Abstract:

**PURPOSE:** To enhance the accuracy of code quantity control without increasing the scale of hardware by shifting a cross reference characteristic between a code quantity and a scale factor based on the number of components of picture data.

**CONSTITUTION:** The number of components for an input picture is added to a compression rate as a parameter for range setting of a scale factor  $\alpha_i$  of an MPU 33 to be multiplied by a quantization table 32 and the scale factor  $\alpha_i$  is shifted depending on the number of components to decide the scale factor  $\alpha_i$  corresponding to the code quantity and to implement quantization. Furthermore, a byte stuffing generated quantity is added to each output of code quantity calculation circuits 36-39 by adder circuits 40-43 to obtain a code quantity of each group corresponding to each scale factor and a scale factor  $\alpha_t$  corresponding to a set code quantity  $N_t$  is calculated by a calculation circuit 46 based on the relation with each scale factor to prevent a set value overflow of code quantity by the byte stuffing processing.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

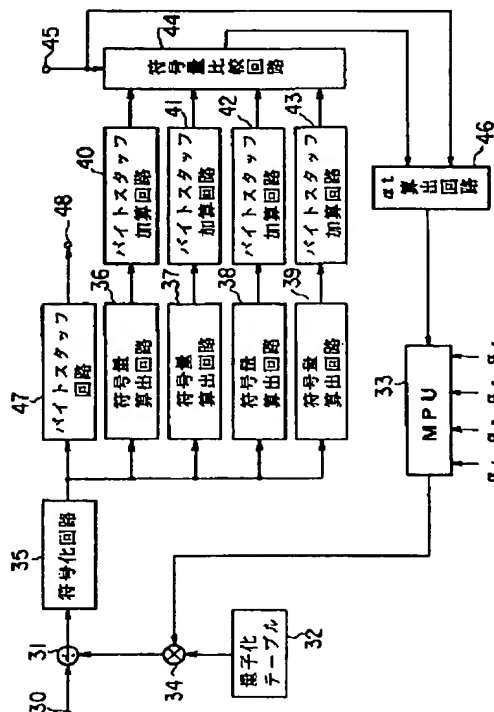
[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(11)特許出願公開番号



## 【特許請求の範囲】

【請求項1】 画像データにそれぞれが複数の画素でなるブロック単位で直交変換処理を施す直交変換手段と、この直交変換手段から出力される変換係数をブロック単位で複数のグループに分類し、予め用意された量子化テーブルにスケールファクタを乗算した値で各グループ毎に変換係数を除算することにより、量子化処理を施す量子化手段と、この量子化手段から出力される変換係数にブロック単位で符号化処理を施す符号化手段と、この符号化手段から出力される符号化データのグループ毎の符号量をそれぞれ算出する符号量算出手段と、この符号量算出手段で算出されたグループ毎の符号量と予め設定された設定符号量との比較結果に基づいて、符号量とスケールファクタとの対応特性から前記量子化テーブルに乘算する前記スケールファクタを生成する生成手段と、前記画像データのコンポーネント数に基づいて、前記符号量とスケールファクタとの対応特性をシフトさせて前記生成手段に供させる制御手段とを具備してなることを特徴とする画像圧縮符号化装置。

【請求項2】 画像データにそれぞれが複数の画素でなるブロック単位で直交変換処理を施す直交変換手段と、この直交変換手段から出力される変換係数をブロック単位で複数のグループに分類し、予め用意された量子化テーブルにスケールファクタを乗算した値で各グループ毎に変換係数を除算することにより、量子化処理を施す量子化手段と、この量子化手段から出力される変換係数にブロック単位で符号化処理を施す符号化手段と、この符号化手段から出力される符号化データのグループ毎の符号量をそれぞれ算出する符号量算出手段と、この符号量算出手段で算出されたグループ毎の符号量に、バイトスタッキングの発生量とスケールファクタとの対応特性から求められるバイトスタッキング発生量をそれぞれ加算する加算手段と、この加算手段から出力されるグループ毎の符号量と予め設定された設定符号量との比較結果に基づいて、符号量とスケールファクタとの対応特性から前記量子化テーブルに乘算する前記スケールファクタを生成する生成手段とを具備してなることを特徴とする画像圧縮符号化装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 この発明は、デジタル化された映像信号に圧縮符号化処理を施す画像圧縮符号化装置の改良に関する。

## 【0002】

【従来の技術】 周知のように、近年では、映像機器のデジタル化に伴ない、カメラにおける信号処理もデジタル化が促進されている。これに付随して、デジタル化した映像信号をデジタルのまま記録媒体に記録するようなシステムの開発も盛んに行なわれている。このシステムの一例として、電子スチルカメラがあげられる。この電子

スチルカメラとしては、フロッピーディスクに映像信号をアナログ記録するものが既に市販されているが、最近では、メモ리카ードに映像信号をデジタル記録するものが発表されている。この場合、メモ리카ードの限りあるメモリ容量を効率よく利用するために、画像データの圧縮技術が必須となっている。

【0003】 既に発表されている画像データの圧縮手段としては、DPCM (DefferencialPulse Code Modulation) 方式や直交変換方式等があるが、現在のところ、圧縮率を高めようとする動きが強いため、直交変換方式の1つであるDCT (Discrete Cosine Transform) を用いたデータ圧縮が有利であると言われている。また、静止画像の圧縮符号化方式の国際標準化が、ISO/CCTTの下部組織であるJPEG (Joint Photographic Expert Group) によってなされ、その中で一番利用されるであろうベースライン方式も、DCTを使用した圧縮符号化方式となっている。

【0004】 ところで、画像を圧縮符号化した場合に、一般的に細かい絵柄に対しては符号量が増大し、逆に滑らかな画像では符号量が減少する方向にある。このため、画像毎に圧縮時の符号量が変化するため、記録媒体に記録する際の記録容量の管理が困難になるという問題が生じる。上述したベースライン方式では、この点に関して何も定義されていないため、実際の運用では量子化時に用いる量子化テーブルの値を加減することで、最終的な符号量を一定にするためのいわゆる符号量制御を施すことが一般的になっている。

【0005】 ただし、この場合、ハードウェア規模を考慮すると、システムが持つことのできる量子化テーブルの数に限りがあるため、それぞれの設定符号量に対して代表的なものを数例設定しておくのが一般的である。このため、画像データの内容によっては制御後の符号量の精度に問題を残すものがでてきている。

【0006】 図8は、メモ리카ードを用いた電子スチルカメラに採用されている、従来の画像圧縮符号化装置を示している。なお、図8では、説明を簡略化するために白黒画像の場合について示している。すなわち、入力端子11に供給されたインターレース方式のデジタル画像データは、メモリコントローラ12に入力される。このメモリコントローラ12は、入力されたデジタル画像データをノンインターレース方式のデータに変換するためにフレームメモリ13に一旦記憶させる。その後、メモリコントローラ12は、フレームメモリ13から水平方向8画素×垂直方向8画素でなる64画素を1つのブロックとして、デジタル画像データをブロック毎に読み出し、DCT回路14とエンコーダ15とに出力している。

【0007】 図9は、1つのブロックを示している。図9において、第1ライン～第4ラインは第1フィールドのデータであり、第nライン～第(n+3)ラインは第

10

20

30

40

50

2フィールドのデータである。なお、 $n$ は任意の定数である。また、画素は、第1ラインのX11からX18、次は第 $n$ ラインのX21からX28という順序で読み出され、最後に第 $(n+3)$ ラインのX88が読み出されるようになる。そして、DCT回路14は、入力されたブロック単位のデータに離散コサイン変換処理を施し、その変換係数をエンコーダ15に出力している。エンコーダ15は、入力された変換係数に量子化処理を施した後、ハフマン符号化処理を施すことで可変長符号化処理を行ない、これにより生成された圧縮画像データが、カードコントローラ16を介してメモリカード17に供給される。

【0008】電子スチルカメラの場合、メモリカード17が静止画像を何枚記録することができるか、つまり通常のフィルムにおける何枚撮りかが不定であると使用者にとって不便である。このため、メモリカード17に記録する静止画像1枚分の圧縮画像データの量は、固定であることが必要となる。すなわち、どのような被写体を撮影しても、その撮影された静止画像1枚分に対応する全デジタル画像データを、一定のデータ量(符号量)以下に圧縮して固定長化する必要が生じる。

【0009】ここで、任意の画像データを圧縮して固定長化する場合、画面全体に対してどのように符号量を配分するかが重要な点となる。例えば画面の上部から順番に符号化していくと、途中で定められた符号量に達してしまい、画面の下部を符号化することができなくなることもある。このため、ある程度均一でしかも画質を損なわないように、画面全体に符号量を適切に配分して符号化する必要がある。

【0010】一般に、絵柄の変化がほとんどないブロックは、ほぼDC(直流)成分しかなく、逆に絵柄の変化が激しいブロックはDC成分の他にAC(交流)成分が\*

$$Bact = \sum_{i=1}^8 \left\{ \sum_{j=1}^6 |X(i, j) - X(i, j+2)| \right\} + \sum_{i=1}^6 \left\{ \sum_{j=1}^8 |X(i, j) - X(i+2, j)| \right\} \dots (1)$$

【0014】さらに、アクティビティ算出回路19は、(1)式で算出した各ブロック毎のアクティビティBactを、1フレーム分加算することでフレームアクティビティを算出し、ブロックビット配分回路20と $\alpha$ 算出回路21とに出力している。このうち、 $\alpha$ 算出回路21は、入力されたフレームアクティビティに基づいて、量子化テーブル22の出力を正規化するための係数 $\alpha$ を算出している。

【0015】次に、2スキャン目でDCT及び符号化処理が行なわれる。すなわち、上記DCT回路14の出力は、入力端子23を介して除算回路24に供給される。

\*多く存在している。このため、圧縮符号化を行なう際に、AC成分を削ってしまうと画質の歪みが大きくなる。そこで、周波数変化量の少ない(絵柄の変化が少ない)ブロックには符号量を少なめに割り振り、周波数変化量の多い(絵柄の変化の多い)ブロックには符号量を多めに割り振るように圧縮符号化を行なうことによって、画質の劣化が分散され歪みを目立たなくすることができる。

【0011】したがって、上記エンコーダ15においては周波数変化量を情報量ととらえ、各ブロック毎の情報量の多さ(アクティビティ)を求め、このアクティビティに基づいて各ブロック毎に符号化できる符号量を制御することで固定長化を行なうようにしている。なお、エンコーダ15では、固定長化を実現するために2スキャン方式を採用している。この2スキャン方式は、1スキャン目に画像1枚のアクティビティの総量であるフレームアクティビティを検出し、このフレームアクティビティに基づいて2スキャン目で圧縮符号化を行なうことで、固定長化を実現している。

【0012】図10は、上記エンコーダ15の構成を示している。まず、1スキャン目にフレームメモリ13から読み出されたブロック毎のデータは、入力端子18を介してアクティビティ算出回路19に供給される。アクティビティ算出回路19は、入力されたブロック毎のデータに対して(1)式に示す演算を行なうことで、各ブロック毎のアクティビティBactを算出している。この式は、垂直及び水平のBPF(バンドパスフィルタ)出力の総和を求めており、式中 $X(i, j)$ は図9に示した $X_{ij}$ に対応している。

【0013】  
【数1】

また、 $\alpha$ 算出回路21で算出された係数 $\alpha$ と量子化テーブル22の出力とを乗算回路25で乗算した値も、除算回路24に供給される。そして、除算回路24によりDCT回路14の出力を乗算回路25の出力で割ることによって、DCT回路14の出力である変換係数の振幅制御が行なわれ符号化回路26に供給される。符号化回路26では、入力された変換係数にハフマン符号化処理を施し、得られた可変長符号化データを符号量制御回路27に出力している。

【0016】ここで、上記ブロックビット配分回路20は、アクティビティ算出回路19で算出されたブロック

アクティビティ Bact と、入力端子 28 を介して供給される 1 フレーム当たりの目標とする設定符号量と、1 スキャン目で算出されたフレームアクティビティとに基づいて、各ブロック毎の符号のビット配分量を設定し、符号量制御回路 27 に出力している。このため、符号量制御回路 27 は、設定されたビット配分量を越えないように符号化データの符号量を制御することで符号量を固定長化し、出力端子 29 に出力している。

【0017】前述したように、上記のような従来の画像圧縮符号化装置では、システムが持つことの可能な量子化テーブルの数に限りがあるため、画像データの内容によっては、設定している量子化テーブルでは符号量制御後の符号量の精度に問題がある場合が生じている。

【0018】また、先に述べた J P E G のベースライン方式の規格においては、バイト化した符号化データ中に F F (16 進) が現われた場合に、その直後に 0 0 (16 進) データを新たに追加挿入するバイトスタッフィングという規則が設けられている。すなわち、符号化データを含む J P E G ベースライン方式のデータ構造の中では、管理コード (マーカコード) が定義されており、この管理コードが F F, X X (16 進、X X は 0 0 以外の定められた値) という 2 バイトで定められている。J P E G に準拠するシステムは、このマーカコードに基づいて符号化データを管理するが、このときにこのマーカコードと実際の符号化データに偶然現われる F F, X X (16 進) とを混同しないように、バイトスタッフィング操作が行なわれる。

【0019】このため、J P E G のベースライン方式に準拠した圧縮符号化処理を、符号量制御しながら行なう場合、設定符号量以内に符号化しようとしても、バイトスタッフィング操作により F F (16 進) の発生量分の符号量が増える、つまり本来の符号化データ以外のデータが挿入されることになるので、設定符号量を越えてしまう可能性があり、符号量制御を行なう際に大きな障害になるという問題も発生している。

#### 【0020】

【発明が解決しようとする課題】以上のように、従来の画像圧縮符号化装置では、システムが持つことの可能な量子化テーブルの数に限りがあるため、画像データの内容によっては、設定している量子化テーブルでは符号量制御後の符号量の精度に問題が生じる場合がある。また、バイトスタッフィングがある場合には、本来の符号化データ以外のデータが挿入されることになるので、設定符号量を越えてしまう可能性があり、符号量制御を行なう際に大きな障害になるという問題を有している。

【0021】そこで、この発明は上記事情を考慮してなされたもので、ハードウェアの規模を大幅に増やすことなく精度の高い符号量制御を行ない得る極めて良好な画像圧縮符号化装置を提供することを目的とする。また、この発明は、バイトスタッフィングを考慮した符号量制

御を行ない得る極めて良好な画像圧縮符号化装置を提供することを目的とする。

#### 【0022】

【課題を解決するための手段】この発明に係る画像圧縮符号化装置は、画像データにそれぞれが複数の画素でなるブロック単位で直交変換処理を施す直交変換手段と、この直交変換手段から出力される変換係数をブロック単位で複数のグループに分類し、予め用意された量子化テーブルにスケールファクタを乗算した値で各グループ毎に変換係数を除算することにより、量子化処理を施す量子化手段と、この量子化手段から出力される変換係数にブロック単位で符号化処理を施す符号化手段と、この符号化手段から出力される符号化データのグループ毎の符号量をそれぞれ算出する符号量算出手段と、この符号量算出手段で算出されたグループ毎の符号量と予め設定された設定符号量との比較結果に基づいて、符号量とスケールファクタとの対応特性から量子化テーブルに乘算するスケールファクタを生成する生成手段と、画像データのコンポーネント数に基づいて、符号量とスケールファクタとの対応特性をシフトさせて生成手段に供させる制御手段とを備えるようにしたものである。

【0023】また、この発明に係る画像圧縮符号化装置は、画像データにそれぞれが複数の画素でなるブロック単位で直交変換処理を施す直交変換手段と、この直交変換手段から出力される変換係数をブロック単位で複数のグループに分類し、予め用意された量子化テーブルにスケールファクタを乗算した値で各グループ毎に変換係数を除算することにより、量子化処理を施す量子化手段と、この量子化手段から出力される変換係数にブロック単位で符号化処理を施す符号化手段と、この符号化手段から出力される符号化データのグループ毎の符号量をそれぞれ算出する符号量算出手段と、この符号量算出手段で算出されたグループ毎の符号量に、バイトスタッフィングの発生量とスケールファクタとの対応特性から求められるバイトスタッフィング発生量をそれぞれ加算する加算手段と、この加算手段から出力されるグループ毎の符号量と予め設定された設定符号量との比較結果に基づいて、符号量とスケールファクタとの対応特性から量子化テーブルに乘算するスケールファクタを生成する生成手段とを備えるようにしたものである。

#### 【0024】

【作用】上記のような構成によれば、まず、グループ毎の符号量と予め設定された設定符号量との比較結果に基づいて、符号量とスケールファクタとの対応特性から量子化テーブルに乘算するスケールファクタを生成する生成手段に対して、画像データのコンポーネント数に基づいて、符号量とスケールファクタとの対応特性をシフトさせるようにしたので、ハードウェアの規模を大幅に増やすことなく精度の高い符号量制御を行なうことができるようになる。

【0025】また、グループ毎の符号量に、バイトスタフピングの発生量とスケールファクタとの対応特性から求められるバイトスタフピング発生量をそれぞれ加算し、その加算結果と予め設定された設定符号量との比較結果に基づいて、符号量とスケールファクタとの対応特性から量子化テーブルに乘算するスケールファクタを生成するようにしたので、バイトスタフピング処理を行なっても符号量が設定値をオーバーすることを防止することができ、バイトスタフピングを考慮した符号量制御を行なうことができる。

#### 【0026】

【実施例】以下、この発明の一実施例について図面を参照して詳細に説明する。図1において、符号30は入力端子である。この入力端子30には、DCT回路14から出力されるDCT変換係数が供給される。すなわち、先に述べたように、図8に示したフレームメモリ13に記録されたデジタル画像データは、2回スキャンされる。第1回目のスキャンでは、以下に説明する符号量の計算とスケールファクタ（係数） $\alpha_t$ の計算とが行なわれ、第2回目のスキャンでは、算出されたスケールファクタ $\alpha_t$ を用いて実際の符号化が行なわれる。

【0027】まず、第1回目のスキャンでフレームメモリ13から読み出されたデジタル画像データが、DCT回路14に供給されてDCT変換係数が生成され、このDCT変換係数が入力端子30に供給される。この場合、DCT回路14では、生成したDCT変換係数の各ブロックを、図2に示すようにブロック単位で4つのグループG1、G2、G3、G4に分類し、各グループG1、G2、G3、G4を順次1画像分入力端子30に出力している。

【0028】この入力端子30に供給されたDCT変換係数は、除算回路31に供給され、量子化テーブル32とMPU（マイクロプロセッサユニット）33から出力されるスケールファクタとを乗算回路34で乗算した値で除算される。このとき、乗算回路34に供給されるスケールファクタとしては、入力端子30に入力されるブロックのグループG1、G2、G3、G4に同期して、MPU33から $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$ 、 $\alpha_4$ が与えられる。すなわち、グループGi（ $i=1, 2, 3, 4$ ）は、それぞれに対応するスケールファクタ $\alpha_i$ （ $i=1, 2, 3, 4$ ）を用いて除算回路31で量子化されることになる。

【0029】除算回路31で量子化されたDCT変換係数は、符号化回路35に供給されブロック単位で可変長符号化処理が施された後、各グループG1、G2、G3、G4に対応して設置された符号量算出回路36、37、38、39にそれぞれ供給されて各グループ毎に積算される。ここでは、1画像データがG1、G2、G3、G4の4つのグループに分けられているため、符号量算出回路36～39から出力される各積算値は、1画

像データの符号積算量の約 $1/4$ となっている。このため、符号量算出回路36～39では、積算値を4倍して符号量N1、N2、N3、N4を得ている。

【0030】図3は、上記スケールファクタ $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$ 、 $\alpha_4$ と符号量N1、N2、N3、N4との関係を示している。今、設定符号量Ntが、図3に示す位置にあるとすると、A（ $\alpha_2$ 、N2）点とB（ $\alpha_3$ 、N3）点との2点で直線近似した直線式から符号量Ntに対応するスケールファクタ $\alpha_t$ が算出される。このため、各符号量算出回路36、37、38、39から出力される符号量N1、N2、N3、N4を、それぞれ後述するバイトスタフピング量加算回路40、41、42、43を介して符号量比較回路44に供給し、入力端子45に供給された設定符号量Ntに最も近い値と2番目に近い値とを選択する。そして、その選択された2つの符号量を $\alpha_t$ 算出回路46に供給して2点直線近似した直線式より、設定符号量Ntに対応するスケールファクタ $\alpha_t$ を算出している。

【0031】ここで、スケールファクタ $\alpha_i$ の範囲について注目すると、この符号化アルゴリズムにおいて画像を符号化した場合、仮に量子化変換係数が予め設定していたスケールファクタ $\alpha_i$ の範囲に入らない（越えるかまたは下回る）場合、量子化時のスケールファクタ $\alpha_t$ は強制的に $\alpha_i$ の最大値または最小値に固定されるため、量子化後の符号量は設定しようとする目標値を大幅に越えるまたは足りない可能性が出てくる。これを防ぐためには、予め設定する $\alpha_i$ の範囲の組み合わせを増やしてやればよいが、これは現実問題としてハードウェア規模の増大に結び付くため、むやみに増やすわけにはいかない。

【0032】一般的に、圧縮符号化されるために入力される画像データにおいて、圧縮符号化時に重要な画像データの内容を示すパラメータは、その画像の持つ情報量である。画像の細かさは前述したDCT変換係数にて表現されるが、それ以外のパラメータで重要なものに、画像を構成するコンポーネントの数がある。このコンポーネント数で表現することができる内容としては、例えば入力画像がカラーか白黒かということがある。画像の情報量からみると、同一画像、同一画素数のもので、白黒画像はカラー画像の約半分の情報量しかない。

【0033】この場合、上述した圧縮符号化方式を採用すると、 $\alpha_i$ の設定範囲をカラー画像を想定して行なった場合、その範囲を白黒画像に適用すれば、元の情報量の差から $\alpha_i$ を最小にしても最終的な量子化時の $\alpha_t$ は大きすぎることになり、最終的な符号量は目標に対して下回ることになる。また、 $\alpha_i$ の範囲設定を白黒画像で行なった場合は、全く逆の結果となってしまう、目標符号量に対してオーバーしてしまうことになる。

【0034】そこで、この実施例では、 $\alpha_i$ の範囲設定を行なうパラメータとして、圧縮率に加え入力画像構成

10

20

30

40

50

コンポーネント数も加えるようにしている。具体的には、図4に示すように、デフォルトとして入力コンポーネント数が3（カラー画像を想定）で $\alpha 1 \sim \alpha 4$ までの設定を行ない、これを元にしてコンポーネント数が少なくなれば図4中左側にシフトした $\alpha 1' \sim \alpha 4'$ を設定し、逆にコンポーネント数が多い場合は図4中右側にシフトした $\alpha 1'' \sim \alpha 4''$ を設定して、それぞれ $\alpha t$ を決定して量子化を行なうようにしている。

【0035】次に、前述したように、J P E G対応においては、バイトスタッキングによるデータ量の増分がある。図5は、バイトスタッキングの発生量とスケールファクタ $\alpha$ との関係を示している。すなわち、任意の画像をある設定符号量（例えば2 b i t / p e l）に最も近付くようにスケールファクタ $\alpha$ を決めて符号化すると、バイトスタッキングの発生量は図5に示す曲線の図中下側の部分に分布する。

【0036】この実施例では、符号量算出回路36～39の各出力をバイトスタッキング量加算回路40～43に供給している。バイトスタッキング量加算回路40～43では、符号量算出回路36～39の各出力に、図5に示したテーブルにしたがってバイトスタッキング発生量 $B 1, B 2, B 3, B 4$ を加算することで、スケールファクタ $\alpha 1, \alpha 2, \alpha 3, \alpha 4$ に対応した各グループの符号量 $N 1+B 1, N 2+B 2, N 3+B 3, N 4+B 4$ を得る。

【0037】これら得られた符号量 $N 1+B 1, N 2+B 2, N 3+B 3, N 4+B 4$ を、符号量比較回路44で入力端子45に供給された設定符号量 $N t$ と比較して、設定符号量 $N t$ に最も近い値と2番目に近い値とを選択する。つまり前述したように $A'(\alpha 2, N 2+B 2)$ 点と $B(\alpha 3, N 3+B 3)$ 点との2点を求める。そして、その選択された2つの符号量を $\alpha t$ 算出回路46に供給して2点直線近似した直線式より、設定符号量 $N t$ に対応するスケールファクタ $\alpha t$ を算出している。

【0038】次に、第2回目のスキャンでは、M P U 3でスケールファクタ $\alpha 1, \alpha 2, \alpha 3, \alpha 4$ を選択するように設定して、再び入力端子30に図2に示した画像の全ブロックを順次入力し、それらを除算回路31で量子化し符号化回路35で可変長符号化した後、バイトスタッキング回路47でバイト化及びF F（16進）データが発生した場合バイトスタッキング処理【00（16進）の挿入】を行なって、出力端子48から出力される。

【0039】したがって、上記実施例のような構成によれば、まず、量子化テーブル32に乗算すべきスケールファクタ $\alpha i$ の範囲設定を行なうパラメータとして、圧縮率に加えて入力画像を構成するコンポーネント数も加えるようにし、例えばコンポーネント数が少なくなればスケールファクタ $\alpha i$ を図4中左側にシフトした $\alpha 1' \sim \alpha 4'$ を設定し、逆にコンポーネント数が多い場合は

図4中右側にシフトした $\alpha 1'' \sim \alpha 4''$ を設定して、それぞれ $\alpha t$ を決定して量子化を行なうようにしたので、ハードウェアの規模を大幅に増やすことなく精度の高い符号量制御を行なうことができるようになる。

【0040】また、符号量算出回路36～39の各出力に、図5に示したテーブルにしたがってバイトスタッキング発生量 $B 1, B 2, B 3, B 4$ を加算することで、スケールファクタ $\alpha 1, \alpha 2, \alpha 3, \alpha 4$ に対応した各グループの符号量 $N 1+B 1, N 2+B 2, N 3+B 3, N 4+B 4$ をそれぞれ得、これら得られた符号量 $N 1+B 1, N 2+B 2, N 3+B 3, N 4+B 4$ とスケールファクタ $\alpha 1, \alpha 2, \alpha 3, \alpha 4$ との関係から、設定符号量 $N t$ に対応するスケールファクタ $\alpha t$ を算出するようにしたので、バイトスタッキング処理を行なっても符号量が設定値をオーバーすることを防止することができ、バイトスタッキングを考慮したJ P E G準拠の符号量制御を行なうことができる。

【0041】次に、図6は、この発明の他の実施例を示している。図10と同一部分には同一符号を付して説明すると、まず、前述したバイトスタッキングの発生量とフレームアクティビティとの間には、図7に示す関係がある。すなわち、任意の画像をある設定符号量（例えば2 b i t / p e l）に最も近付くように符号化すると、バイトスタッキングの発生量は図7に斜線で示す部分に分布する。

【0042】そこで、図6に示す実施例では、アクティビティ算出回路19から出力されるフレームアクティビティをバイトスタッキング量推定テーブル49に供給することにより、フレームアクティビティから推定されるバイトスタッキングの発生量を求める。そして、この求めたバイトスタッキングの発生量を、減算回路50により、入力端子28に供給された設定符号量から減算した値でブロックビット配分回路20を制御することにより、バイトスタッキングを考慮した符号量制御を行なうようにしている。

【0043】また、他の例としては、ブロックアクティビティとして、入力端子18にD C T変換係数を入力し、そのA C成分の絶対値和をとるようにしてもよい。なお、この発明は上記各実施例に限定されるものではなく、この外その要旨を逸脱しない範囲で種々変形して実施することができる。

【0044】

【発明の効果】以上詳述したようにこの発明によれば、ハードウェアの規模を大幅に増やすことなく精度の高い符号量制御を行ない得る極めて良好な画像圧縮符号化装置を提供することができる。また、この発明によれば、バイトスタッキングを考慮した符号量制御を行ない得る極めて良好な画像圧縮符号化装置を提供することができる。

【図面の簡単な説明】

【図1】この発明に係る画像圧縮符号化装置の一実施例を示すブロック構成図。

【図2】DCT変換係数のグループ分けを説明するために示す図。

【図3】符号量とスケールファクタとの関係を示す図。

【図4】同実施例の要部を説明するために示す図。

【図5】バイトスタフing発生量とスケールファクタとの関係を示す図。

【図6】この発明の他の実施例を示すブロック構成図。

【図7】バイトスタフing発生量とフレームアクティビティとの関係を示す図。

【図8】従来の画像圧縮符号化装置を示すブロック構成図。

【図9】水平方向8画素×垂直方向8画素でなる1つのブロック構成を示す図。

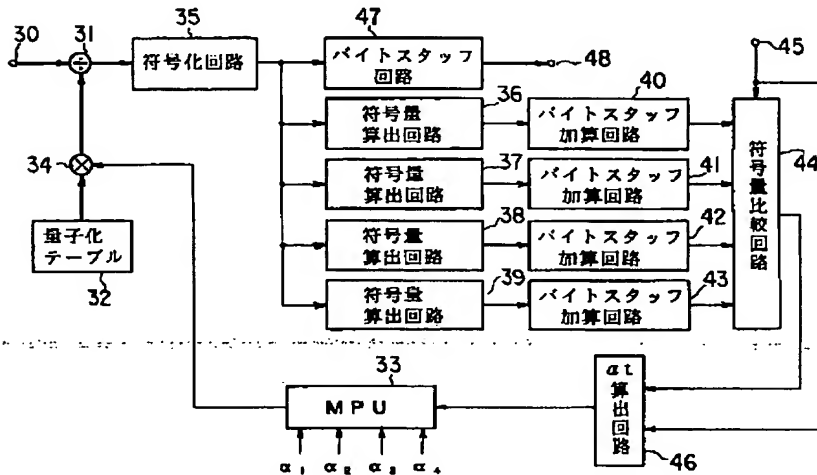
【図10】同従来装置のエンコーダを示すブロック構成図。

\* 【符号の説明】

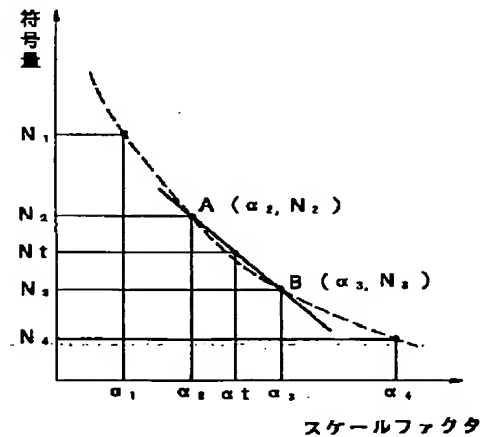
11…入力端子、12…メモリコントローラ、13…フレームメモリ、14…DCT回路、15…エンコーダ、16…カードコントローラ、17…メモリカード、18…入力端子、19…アクティビティ算出回路、20…ブロックビット配分回路、21… $\alpha$ 算出回路、22…量子化テーブル、23…入力端子、24…除算回路、25…乗算回路、26…符号化回路、27…符号量制御回路、28…入力端子、29…出力端子、30…入力端子、31…除算回路、32…量子化テーブル、33…MPU、34…乗算回路、35…符号化回路、36～39…符号量算出回路、40～43…バイトスタフing量加算回路、44…符号量比較回路、45…入力端子、46… $\alpha$ t算出回路、47…バイトスタフing回路、48…出力端子、49…バイトスタフing量推定テーブル、50…減算回路。

\*

【図1】



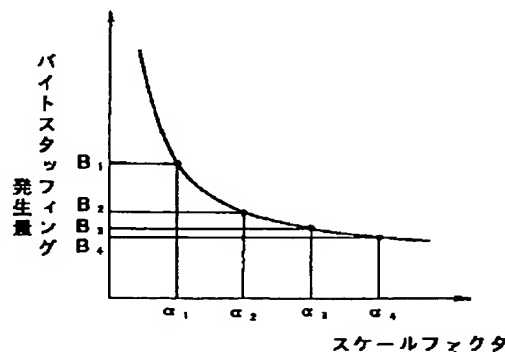
【図3】



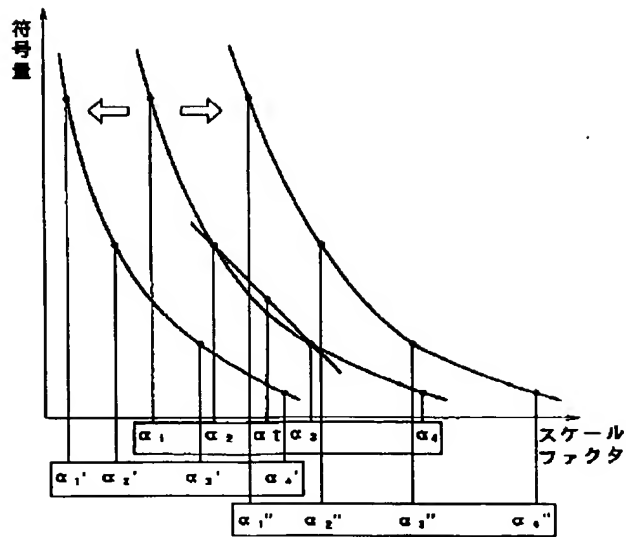
【図2】

G1	G2	G3	G4	G1	G2	G3	G4	-----
G1	G2	G3	G4	G1	G2	G3	G4	-----
G1	G2	G3	G4	G1	G2	G3	G4	-----
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

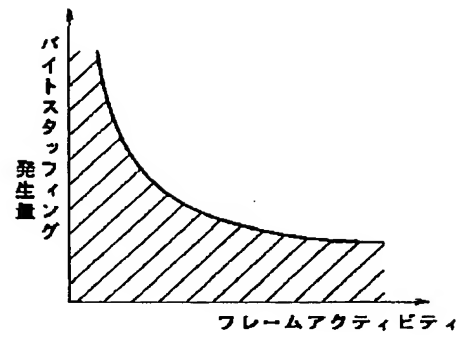
【図5】



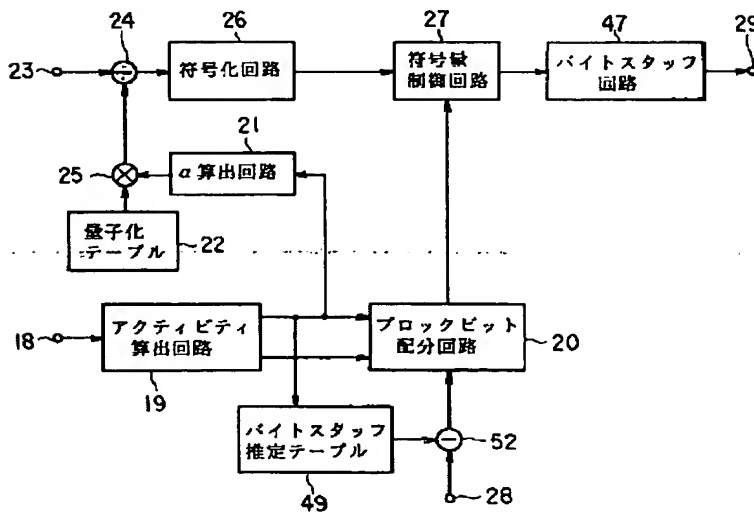
【図4】



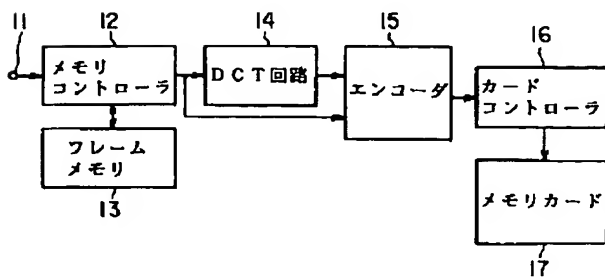
【図7】



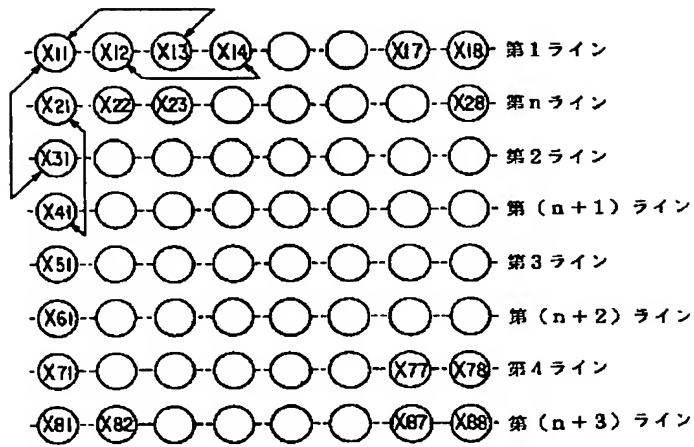
【図6】



【図8】



【図9】



【図10】

